Ewaybot\_UI 说明文档

版本控制

|  |  |  |
| --- | --- | --- |
| 修改日期 | 说明 | 修订人 |
| 2015-10-21 | 文档建立 | 史皓天 |

类名：RobotControl

描述：网络端UI 向上位机发送指令的python接口

私有变量描述：

1. arm\_control\_xy 定义手臂运动控制 W A S D STOP 键位对应的控制方法

2. armBuildInAction 定义手臂运动 预设动作 INIT SWING WAVE STOP 键位对应的控制方法

3. camControl 定义摄像机开关键位 ON OFF 键位对应的控制方法

4. data\_additional UI系统控制指令

类型：int

数据说明： 0，NULL；1，camera on ；2，camera off

5. z\_pos 记录手臂z方向位置信息

类型：int

数据说明： 初始位默认50，最大位置为100，最小位置为0

6. command\_flag 信号端口指示位

类型：bool

数据说明：False，无新指令；True，有新指令

7. command\_data\_wheels 轮子控制信号

类型：int[6]

数据说明：[angle\_L speed\_L status\_L angle\_R speed\_R status\_R]

L:左轮 R:右轮

angle:转动角度 区间为 -90～+90

speed:转动速度 区间为0～255

status：0 断电 1 刹车 2 反转 3 正转

8. command\_data\_arms 手臂控制信号

类型：int[7]

数据说明：[FLAG v1\_L v2\_L v3\_L v1\_R v2\_R v3\_R]

L:左轮 R:右轮

v1\_L:v3\_L = 左手数据

v1\_R:v3\_R = 左手数据

FLAG 0,Null;1,初始位置；2，Cartesian Move；3，Swing

4，wave; 5,初始位置2

…

外部接口描述：

1. start\_ui\_system(\*args) 启动UI系统

2. clean\_command\_flag() command\_flag置 flase

3. clean\_command\_arms() command\_data\_arms=[0,0,0,0,0,0,0] 数据归零

4. clean\_data\_additional() data\_additional = 0 数据归零 command\_flag置 false

5. output\_command() 输出数据

功能示例：

1.启动UI系统

port = 3000 #端口信息

server = '139.129.130.13'

UI = RoboticControl() #实例化对象

UI.start\_ui\_system([server,port]) #开启UI系统

2.输出数据

UI = RoboticControl()

flag,data\_wheels, data\_arms, data\_additional = UI.output\_command()

注意事项：

1. 每次完成UI端信号读取后，需要使用 clean\_command\_arms 来清理数据

2. 如果data\_additional有数据（>0）,则需要使用 clean\_data\_additional方法 清理数据

参考代码

if source == UI :

flag\_UI, data\_w, data\_a, data\_additional= UI.output\_command()

if data\_additional == 2:

start\_local\_vision\_system()

UI.clean\_data\_additional() #注意事项2

UI.clean\_command\_arms #注意事项1

3. 机器人坐标系定义

|z y

| /

| /

| /

| /

|/\_ \_ \_ \_ \_ x

如图所示，x-y-z组成右手坐标系，y方向重合于机器人前端朝向，z方向垂直于地面方向，x方向垂直于y 轴和z轴。

具体函数定义：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Wheels\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名：on\_move（\*args）

描述：讲摇杆参数转化为指令信号

输入参数定义：直接从socket端读取

输出参数定义：无输出，私有变量 command\_flag = True， command\_data\_wheels 相应变化

调用实例：

robot\_ui= RobotControl() #实例化对象

socketIO.on('move', robot\_ui.on\_move) #启动监听摇杆指令

函数名：\_button\_forward（speed,angle）

描述：输出前进指令，调用于 on\_move

输入参数定义： speed:速度 angle：角度

输出参数定义：data = [60, 150, 3, 60, 150, 3]

调用实例：

command\_data\_wheels = \_button\_forward(speed, angle)

函数名：\_button\_backward（speed,angle）

描述：输出后退指令，调用于 on\_move

输入参数定义：speed:速度 angle：角度

输出参数定义：data = [60, 150, 2, 60, 150, 2]

调用实例：

command\_data\_wheels = \_button\_backward(speed, angle)

函数名：\_button\_turn\_left(speed,angle）

描述：输出左转指令，调用于 on\_move

输入参数定义：speed:速度 angle：角度

输出参数定义：data = [60, 100, 3, 60, 100, 2]

调用实例：

command\_data\_wheels = \_button\_turn\_left(speed, angle)

函数名：\_button\_turn\_right（speed,angle）

描述：输出右转指令，调用于 on\_move

输入参数定义：speed:速度 angle：角度

输出参数定义：data = [60, 100, 2, 60, 100, 3]

调用实例：

command\_data\_wheels = \_button\_turn\_right(speed, angle)

函数名：button\_stop（\*args）

描述：摇杆归0后，发出指令信号

输入参数定义：直接从socket端读取

输出参数定义： 无输出，私有变量 command\_data\_wheels = [0, 0, 0, 0, 0, 0],

command\_flag=false

调用实例：

robot\_ui= RobotControl() #实例化对象

socketIO.on('stop', robot\_ui.button\_stop) #启动监听摇杆归零

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*arm built in\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名：arm\_built\_in（\*args）

描述：点击预设手臂动作按钮后，发出运动指令

输入参数定义：直接从socket端读取

输出参数定义： 无输出，私有变量 command\_data\_arms 相应变化,command\_flag=True

调用实例：

robot\_ui= RobotControl() #实例化对象

socketIO.on('armBI',robot\_ui .arm\_built\_in) #启动监听预设手臂动作

函数名：\_button\_arm\_init（）

描述：输出手臂预设初始化位置1，调用于arm\_built\_in\_action

输入参数定义：无

输出参数定义： 无，私有变量 command\_data\_arms = [1, 0, 0, 0, 0, 0, 0]

command\_flag=True

调用实例：

arm\_build\_in\_action = {'INIT': \_button\_arm\_init}

函数名：\_button\_arm\_swing（）

描述：输出手臂预设 摆臂动作，调用于arm\_built\_in\_action

输入参数定义：无

输出参数定义： 无，私有变量 command\_data\_arms = [3, 0, 0, 0, 0, 0, 0]

command\_flag=True

调用实例：

arm\_build\_in\_action = {'swing': \_button\_arm\_swing}

函数名：\_button\_arm\_wave（）

描述：输出手臂预设 招手动作，调用于arm\_built\_in\_action

输入参数定义：无

输出参数定义： 无，私有变量 command\_data\_arms = [4, 0, 0, 0, 0, 0, 0]

command\_flag=True

调用实例：

arm\_build\_in\_action = {'wave': \_button\_arm\_wave}

函数名：\_button\_arm\_int2（）

描述：输出手臂预设初始化位置2，调用于arm\_built\_in\_action

输入参数定义：无

输出参数定义： 无，私有变量 command\_data\_arms = [5, 0, 0, 0, 0, 0, 0]

command\_flag=True

调用实例：

arm\_build\_in\_action = {'INIT2': \_button\_arm\_int2}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*arm user control\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名：arm\_control(\*args)

描述：控制自定义平面手部运动

输入参数定义：从socket中直接传输

输出参数定义：无输出，私有变量 command\_data\_arms 相应变化,command\_flag=True

调用实例：

robot\_ui= RobotControl() #实例化对象

socketIO.on('arm',robot\_ui.arm\_control) #启动手臂平面运动指令监控

函数名：arm\_control\_z(\*args)

描述：控制自定义深度手部运动

输入参数定义：从socket中直接传输

输出参数定义： 无输出，私有变量 command\_data\_arms 相应变化,command\_flag=True

调用实例：

robot\_ui= RobotControl() #实例化对象

socketIO.on('arm\_z',robot\_ui.cam\_control\_z) #启动手臂深度运动指令监控

函数名：\_button\_arm\_upward（）

描述：自定义 手臂 沿 +z 方向 运动，调用于arm\_control

输入参数定义：无

输出参数定义： 无，私有变量 command\_data\_arms = [2, 0, 0, 1, 0, 0, 1]

command\_flag=True

调用实例：

arm\_build\_in\_action = {'w': \_button\_arm\_upward}

函数名：\_button\_arm\_downward（）

描述：自定义 手臂 沿 -z 方向 运动，调用于arm\_control

输入参数定义：无

输出参数定义： 无，私有变量 command\_data\_arms = [2, 0, 0, -1, 0, 0, -1]

command\_flag=True

调用实例：

arm\_build\_in\_action = {'s': \_button\_arm\_downward}

函数名：\_button\_arm\_turn\_right（）

描述：自定义 手臂 沿 x方向 运动，调用于arm\_control

输入参数定义：无

输出参数定义： 无，私有变量 command\_data\_arms = [2, 1, 0, 0, 1, 0, 0]

command\_flag=True

调用实例：

arm\_build\_in\_action = {'d': \_button\_arm\_turn\_right}

函数名：\_button\_arm\_turn\_left（）

描述：自定义 手臂 沿 -x 方向 运动，调用于arm\_control

输入参数定义：无

输出参数定义： 无，私有变量 command\_data\_arms = [2, -1, 0, 0, -1, 0, 0]

command\_flag=True

调用实例：

arm\_build\_in\_action = {'a': \_button\_arm\_turn\_left}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*camera\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名：cam\_control(\*args)

描述：控制摄像机开关

输入参数定义：从socket中直接传输

输出参数定义： 无，私有变量 data\_additonal 相应变化

调用实例：

robot\_ui= RobotControl() #实例化对象

socketIO.on('cam',robot\_ui.cam\_control) #启动摄像机监控

函数名：\_cam\_on()

描述：控制摄像机开启，调用于 cam\_control

输入参数定义：无

输出参数定义： 无，私有变量 data\_additonal = 2，command\_flag = True

调用实例：

robot\_ui= RobotControl() #实例化对象

socketIO.on('cam',robot\_ui.cam\_control) #启动摄像机监控

函数名：\_cam\_off()

描述：控制摄像机关闭， 调用于cam\_control

输入参数定义：无

输出参数定义： 无，私有变量 data\_additonal = 1，command\_flag = True

调用实例：

robot\_ui= RobotControl() #实例化对象

socketIO.on('cam',robot\_ui.cam\_control) #启动摄像机监控